In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown. Please cancel claims 1, 2, 14-19, 23 and 24 without prejudice.

1. (Cancelled)

2. (Cancelled)

3. (Currently Amended) A method ~~for saving dirty data~~ comprising:

~~a. reading data from a first dirty cache line in a plurality of cache lines in a cache, the first dirty cache line corresponding to a first memory location;~~

~~b. determining if the data is corrupt; and~~

~~c. if the data is not corrupt, then:~~

~~i. writing the data to the first memory location;~~

~~ii. marking the cache line available;~~

~~iii. determining at least one duplicate dirty cache line of the first dirty cache line; and~~

~~iv. marking each of the at least one duplicate dirty cache line as an available cache line.~~

reading data from a first dirty cache line in a cache memory;

determining if the data is corrupt;

marking the first cache line invalid if the data is corrupt;

determining if a duplicate cache line exists;

determining if the data within the duplicate cache line is corrupt if the duplicate cache line exists;

writing the data to a first location in memory if the duplicate cache line is not corrupt; and

marking the first dirty cache line available.

4. (Currently Amended) The method of claim 3, further comprising: ~~wherein each of the duplicate dirty cache lines is marked invalid.~~

writing the data to the first memory location if the data is not corrupt;

marking the cache line available;

determining at least one duplicate dirty cache line for the first dirty cache line; and

marking each duplicate dirty cache line as an available cache line.

5. (Currently Amended) The method of claim 3, ~~additionally~~ further comprising ~~reading each of the duplicate dirty cache lines and~~ marking each duplicate dirty cache line invalid if the data within the duplicate dirty cache line is corrupt.

6. (Currently Amended) The method of claim 3, ~~additionally~~ further comprising determining that no duplicate cache lines exist.~~:~~

~~a. if the data on the first dirty cache line is corrupt, then marking the cache line invalid, and determining if a duplicate cache line exists;~~

~~b. if a duplicate cache line exists, then determining if the data is corrupt, and if the data is not corrupt, then:~~

~~i. writing the data to the first memory location; and~~

~~ii. marking the first dirty cache line available; and~~

~~c. if a duplicate cache line exists, and the data is corrupt, then repeating the method~~

~~at (a) until one of the following occurrences:~~

~~i. a duplicate cache line having data that is not corrupt is found; and~~

~~ii. no more duplicate cache lines exist.~~

7. (Currently Amended) The method of claim 6 4, ~~additionally~~ further comprising terminating the method if a duplicate cache line is not found~~, then terminating the method~~.

8. (Currently Amended) A method ~~for saving dirty data~~ comprising:

~~a. reading data from a first cache line in a plurality of cache lines in a cache, the first cache line corresponding to a first memory location;~~

~~b. determining if the first cache line is a clean line;~~

~~c. determining if the data is corrupt; and~~

~~d. if the data is not corrupt, and the first cache line is not a clean line, then:~~

~~i. writing the data to the first memory location;~~

~~ii. marking the first cache line available; and~~

~~iii. determining at least one duplicate dirty cache line of the first dirty cache line and marking each of the at least one duplicate dirty cache line invalid.~~

reading data from a first cache line in a cache memory;

determining if the first cache line is a clean line;

determining if the data in the first cache line is corrupt;

marking the first cache line invalid if the data is corrupt;

determining if a duplicate cache line exists;

determining if the data within the duplicate cache line is corrupt if the duplicate cache line exists;

writing the data to a first location in memory if the duplicate cache line is not corrupt; and

marking the first dirty cache line available.

9. (Currently Amended) The method of claim 8, ~~additionally~~ further comprising ~~if the data is corrupt, and the first cache line is not clean, then~~:

~~a. marking the cache line invalid, and determining if a duplicate cache line exists;~~

~~b. if a duplicate cache line exists, then determining if the data is corrupt, and if the data is not corrupt, then:~~

~~i. writing the data to the first memory location; and~~

~~ii. marking the first cache line available; and~~

~~c. if a duplicate cache line exists, and the data is corrupt, then repeating the method starting at (a) until one of the following occurrences:~~

~~i. a duplicate cache line having data that is not corrupt is found; and~~

ii. no more duplicate cache lines exist.

marking the cache line invalid if the data is corrupt and the first cache line is not clean;

determining if a duplicate cache line exists;

determining if the data is corrupt if a duplicate cache line exists;

writing the data to a first memory location if the data is not corrupt; and

marking the first cache line available; and

10. (Currently Amended) The method of claim 9, ~~additionally~~ further comprising terminating the method if a duplicate cache line is not found~~, then terminating the method~~.

11. (Currently Amended) The method of claim 10, ~~additionally~~ further comprising reading from a second cache line if the data is corrupt, and the first cache line is clean~~, then repeating the method starting at 8(a) until no more associated cache lines exist~~.

12. (Currently Amended) The method of claim 3, wherein each of the duplicate dirty cache lines is marked invalid.

~~A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instructions which, when executed by a processor, cause the processor to perform the following:~~

~~receive an instruction to write data to a memory location in a memory that is cached to a cache;~~

~~write the data to a plurality of cache lines in the cache, the plurality of cache lines written to being duplicate cache lines; and~~

~~mark the duplicate cache lines as dirty.~~

13. (Currently Amended) ~~The machine-readable medium of claim 12, wherein the cache is a multi-way set-associative cache.~~ The method of claim 9, further comprising determining that no duplicate cache lines exist.

14. (Cancelled)

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Cancelled)

19. (Cancelled)

20. (Currently Amended) An apparatus comprising ~~a cache controller to~~:

a cache memory; and

a cache controller, coupled to the cache memory, to receive ~~intercept~~ a request to write data to a location within the cache~~d~~ memory, ~~location; determine an associated set having cache lines, the associated set corresponding to the memory location; and write the data to a plurality of cache lines in the associated set that are available.~~ read data from a first dirty cache line in the cache memory in response to receiving the request, mark the first cache line invalid if the data is corrupt, determine if a duplicate cache line exists, write the data to a first location in a main memory device if the duplicate cache line is not corrupt, and mark the first dirty cache line available.

21. (Currently Amended) The apparatus of claim 20, ~~further comprising invoking~~ wherein the cache controller invokes a replacement policy ~~of the cache~~ to free up cache lines in the associated set if there are no cache lines available.

22. (Currently Amended) ~~A system comprising:~~ The apparatus of claim 20, wherein the cache controller further writes the data to the first memory location if the data is not corrupt, marks the cache line available, determines at least one duplicate dirty cache line for the first dirty cache line, and marks each duplicate dirty cache line as an available cache line.

~~a main memory having at least one memory location that is cached to a cache;~~

~~a cache having a plurality of cache lines to store data; and~~

~~a processor to receive an instruction to write data to a cached memory location;~~

~~a cache controller to determine an associated set corresponding to the memory location, the associated set having a plurality of cache lines, and to write the data to a plurality of available cache lines in the associated set.~~

23. (Cancelled)

24. (Cancelled)